Chapter 8

整然形式にデータを再構成（P187）

整然データとは

- 以下の3原則に反するデータセットは整然（Tidy）ではない
  - 変数がカラムになっている
  - 観察が行になっている
  - 観察ユニットがテーブルになっている

整然ではないデータとは

- カラム名が変数ではなく、値になっている
- 複数の変数が、カラム名に格納されている
- 複数の観察ユニットが同じテーブルに格納されている
- 1つの観察ユニットが複数のテーブルに格納されている

レシピ65 変数値カラム名をstackで整然化（P190）

- 整然データと混乱データの違いを理解するために、簡単なテーブルが整然かどうかを調べる

In [4]:
```
import pandas as pd
state_fruit = pd.read_csv('state_fruit.csv', index_col=0)
state_fruit
```

Out[4]:

	Apple	Orange	Banana
Texas	12	10	40
Arizona	9	7	12
Florida	0	14	190

- 上記は整然ではない
- カラム名が変数の値になっているため
- 実際、このDataFrameには変数名がそもそも存在しない
- 非整然データを整然データに変換するための第一ステップは、変数を確認すること
- このデータセットでは、stateとfruitという2つの変数が考えられる
- また数値データはweightなど意味のある名前をラベルとして付けることもできる

（1）州名がインデックスになる。カラム名がおかしいので `stack` メソッドでカラム名を解除し、インデックスのレベルに直す

`stack()`

In [5]:
```
state_fruit.stack()
```

Out[5]:
```
Texas    Apple     12
         Orange    10
         Banana    40
Arizona  Apple      9
         Orange     7
         Banana    12
```

```
Florida  Apple      0
         Orange    14
         Banana   190
dtype: int64
```

（2）

- MultiIndexのSeriesになった。インデックスは2階層になった。これで整然データになった。
- state,fruit,weightという変数は鉛直方向
- reset_indexで結果をDataFrameにする

reset_index()

In [8]:
```
state_fruit_tidy = state_fruit.stack().reset_index()
state_fruit_tidy
```

Out[8]:

	level_0	level_1	0
0	Texas	Apple	12
1	Texas	Orange	10
2	Texas	Banana	40
3	Arizona	Apple	9
4	Arizona	Orange	7
5	Arizona	Banana	12
6	Florida	Apple	0
7	Florida	Orange	14
8	Florida	Banana	190

（3）カラム名に名前がないため、適切な識別子に置き換える

In [9]:
```
state_fruit_tidy.columns = ['state', 'fruit', 'weight']
state_fruit_tidy
```

Out[9]:

	state	fruit	weight
0	Texas	Apple	12
1	Texas	Orange	10
2	Texas	Banana	40
3	Arizona	Apple	9
4	Arizona	Orange	7
5	Arizona	Banana	12
6	Florida	Apple	0
7	Florida	Orange	14
8	Florida	Banana	190

（4）（3）は以下のやり方でもカラム名を変更できる

rename_axis() ：Seriesのメソッド、reset_indexを使う前にインデックスレベルの名前を設定

In [11]:
```
state_fruit.stack()\
           .rename_axis(['state', 'fruit'])
```

Out[11]:
```
state    fruit
Texas    Apple     12
         Orange    10
         Banana    40
Arizona  Apple      9
         Orange     7
         Banana    12
Florida  Apple      0
         Orange    14
         Banana   190
dtype: int64
```

（5）（4）の続き

In [12]:
```
state_fruit.stack()\
        .rename_axis(['state', 'fruit'])\
        .reset_index(name='weight')
```

Out[12]:

	state	fruit	weight
0	Texas	Apple	12
1	Texas	Orange	10
2	Texas	Banana	40
3	Arizona	Apple	9
4	Arizona	Orange	7
5	Arizona	Banana	12
6	Florida	Apple	0
7	Florida	Orange	14
8	Florida	Banana	190

（補足）

- `stack` をうまく使うコツは、変換したくないカラムすべてをインデックスに置くこと
- 上記のレシピは州をインデックスに置いたが、そうしないとどうなるか

In [14]:
```
state_fruit2 = pd.read_csv('state_fruit2.csv')
state_fruit2
```

Out[14]:

	State	Apple	Orange	Banana
0	Texas	12	10	40
1	Arizona	9	7	12
2	Florida	0	14	190

In [16]:
```
# これをstackすると、州名がインデックスにないため、すべてのカラムが解除される
state_fruit2.stack()
```

Out[16]:
```
0  State      Texas
   Apple         12
   Orange        10
   Banana        40
1  State     Arizona
   Apple          9
   Orange         7
   Banana        12
```

```
2  State     Florida
   Apple         0
   Orange       14
   Banana      190
dtype: object
```

In [19]:

```
# よって正しくはStateをインデックスにセット後、stack・reset_indexする
state_fruit2.set_index('State').stack().reset_index()
```

Out[19]:

	State	level_1	0
0	Texas	Apple	12
1	Texas	Orange	10
2	Texas	Banana	40
3	Arizona	Apple	9
4	Arizona	Orange	7
5	Arizona	Banana	12
6	Florida	Apple	0
7	Florida	Orange	14
8	Florida	Banana	190